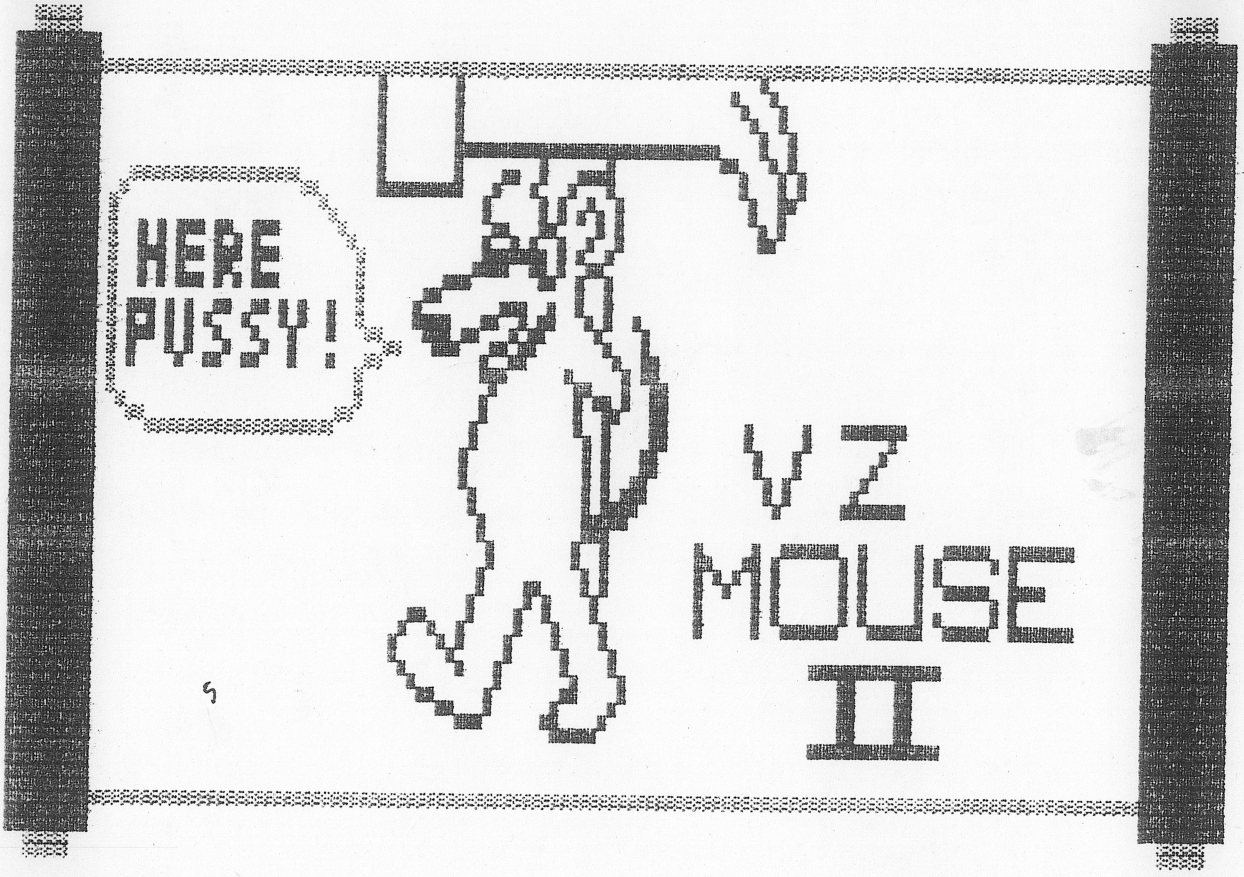
OCTOBER 1992 ISSUE # 40

HUNTER VALLEY VZ JOURNAL

VZ 200 / 300



PRODUCED BI-MONTHLY BY H.V.VZ.U.G.
A NON PROFIT ORGANIZATION

FRONT COVER

JASON OAKLEY HAS BEEN BUSY ON THE FRONT COVER ONCE MORE. AGAIN IT RELATES TO AN ARTICLE INSIDE. THANKS JASON.

HELP – SELL & TELL PAGE 3

VZ BUS MOUSE PART I BY L MILBURN PAGES 4–5
THE FIRST PART OF THE SECOND MOUSE PROJECT FOR THE VZ. LESLIE'S VERSION DIFFERS IN THAT NO MODS ARE REQUIRED TO THE BUS MOUSE, BUT IT'S A BIT MORE EXPENSIVE THAN A SERIAL MOUSE.

KSCAN PT III BY LESLIE MILBURN PAGES 6–7
THE FINAL PART ON KSCAN. DUE TO THE SIZE OF THE SOURCE CODE IT WILL NOT BE PUBLISHED IN THE JOURNAL. READ EDITOR'S COMMENTS ON PAGE 7 FOR MORE DETAILS.

EXTENDED DOS V12.1 BY L MILBURN PPAGES 8–11
THE FIRST PART HAS THE INSTRUCTIONS PLUS HALF THE SOURCE CODE. EXT12.1 IS A VERY USEFUL UTILITY WITH SOME WELCOME FUNCTIONS.

DISKOPS 1 & 2 INSTRUCTIONS PAGE 12
I BELIEVE THIS MAY BE THE FIRST IN A SERIES OF INSTRUCTIONS FOR LESLIE MILBURN'S DISKOP EDITOR ASSEMBLERS.

ASSEMBLY LANGUAGE PART III PAGES 13–15
BY BOB KITCH
PART 3 FEATURES LISTING 5 AND FUNCTIONS IN BASIC.

TECHNICAL DATA SHEET # 1 PAGES 15–16
THE FIRST IN A SERIES OF TECHNICAL DATA SHEETS FOR THE VZ. DISK DRIVE POWER SUPPLY AND WORDPRO CARTRIDGE CIRCUITS ARE PROVIDED.

DRIVE HEAD PARK UPDATE PAGE 17

GARY BULLEY VZ MOUSE ERRATTA PAGE 17
EAGLE EYED GARY HAS SPOTTED SOME OF THE MISTAKES THAT CREPT IN AND PROVIDED CORRECTIONS TO THEM.

TECHNICAL DATA SHEET # 2 PAGE 18
STATIC RAM AND VZ ROM PINOUTS AND BLOCK DIAGRAMS ARE SHOWN.

DAVE MITCHELL SOFTWARE FOR SALE PAGE 19
PATCH3.3 – EXT DOS & MENU/FILE COPIER

PETER HICKMAN SOFTWARE FOR SALE PAGE 19
VZ MODEM & M/C DISASSEMLER

USER GROUPS ** NEWS ** SUBSCRIPTIONS PAGE 20

DISCLAIMER: EVERY EFFORT IS MADE TO INSURE THE ACCURACY OF INFORMATION CONTAINED WITHIN BE IT GENERAL, TECHNICAL, PROGRAMMING, ETC. NO RESPONSIBILITY CAN BE ACCEPTED BY HUNTER VALLEY VZ USERS' GROUP OR AUTHOR AS A RESULT OF APPLYING SUCH INFORMATION IN PRACTICE.

COPYRIGHT: THE HUNTER VALLEY VZ JOURNAL IS SUBJECT TO COPYRIGHT AND NO MATERIAL IN THE JOURNAL MAY BE REPRODUCED IN PART OR WHOLE WITHOUT THE CONSENT OF THE HUNTER VALLEY USERS' GROUP OR THE AUTHOR WHO RETAINS COPYRIGHT.

APOLOGIES:

I STARTED JOURNAL EARLY JULY WHEN THINGS GOT WORSE MEDICALLY. MY CONTINUING MEDICAL PROBELM IS THE RESULT OF THE CAR ACCIDENT OVER 4 YEARS AGO WHICH AFFECTED MY SPINE AND EVERY JOINT IN MY BODY. THE CHRONIC PAINS ARE USUALLY AT A BEARABLE LEVEL, BUT THE WINTER JUST PASSED HAS NOT HELPED.

TO COMPOUND THE PROBLEM, PAIN KILLERS GIVE NO RELIEF AT ALL. THE ONLY THING THAT REDUCES THE CHRONIC PAINS IS LYING DOWN SEVERAL TIMES A DAY WHICH TAKES A BIG CHUNK OUT OF THE DAY. IN THE EVENT OF THE JOURNAL CEASING PRODUCTION OUTSTANDING SUBSCRIPTIONS WOULD BE REFUNDED AS THE CLUB'S FUNDS ARE QUITE SOUND.

IT IS MY INTENTION TO PRUDUCE THE JOURNAL FOR SOME TIME TO COME AS THERE IS STILL A LOT OF LIFE LEFT IN THE HUMBLE VZ. IT GIVES ME A GREAT DEAL OF PLEASURE IN PRUDUCING THE JOURNAL AND BEING PART OF A GROUP OF PEOPLE AUSTRALIA WIDE.

GET WELL SOON NEVILLE:

ON BEHALF OF THE HUNTER VALLEY VZ USER'S GROUP AND MYSELF WE WISH NEVILLE HUGHES A SPEEDY RECOVERY FROM 3 VERY BAD HEART ATTACKS. KEEP UP THE GOOD FIGHT NEVILLE AND WE LOOK FORWARD TO YOUR NEXT PROJECT.

TECHNICAL DATA SHEETS:

I WOULD LIKE YOUR OPINION ON THE DATA SHEETS AS I INTEND THEM TO BECOME A REGULAR ITEM IN THE JOURNAL. IF YOU HAVE A CONTRIBUTION OR IDEAS FOR THEM, THEN PLEASE LET ME KNOW, ED.

FOR SALE - IBM MOTHERBOARD:

A COUPLE WEEKS AGO I INSTALLED AN UP-GRADEABLE 386DX MOTHERBOARD WITH 4M RAM IN MY IBM COMPATIBLE PC. NEEDLESS TO SAY THE OLD MOTHERBOARD IS NO LONGER REQUIRED AND I OFFER IT FOR SALE.

1 OFF BABY AT 286 MOTHERBOARD - 6/12 MHZ
1 MEG RAM AND MATHS CO-PROCESSOR ON BOARD
RAM EXPANDABLE TO 4M USING 4 X 1M SIPPS
1 OFF INSTRUCTION MANUAL

PRICE: $150.00 ONO- INCLUDES POST & PACKING - CONTACT:
JOSEPH P LEON - PHONE (049) 692-399 - 33 TIGHES TCE TIGHES HILL 2297

NEXT ISSUE:

IT IS MY INTENTION TO PRODUCE ANOTHER ISSUE AND HAVE IT POSTED OUT BEFORE CHRISTMAS 1992 AND HOPEFULLY KEEP TO A BI-MONTHLY SCHEDULE. BELOW ARE SOME OF THE PROPOSED MATERIAL FOR NEXT ISSUE.

1) LISTING OF PUBLIC DOMAIN PROGRAMS.
2) LISTING OF PUBLIC DOMAIN SOURCE CODE.
3) ARTICLE ON EDITOR/ASSEMBLER INCOMPATIBILITY.
4) TECHNICAL DATA SHEET # 4, (5-6?).
5) PLUS CONTINUATION OR CONCLUSION OF PARTLY PUBLISHED MATERIAL.

VZ BUS MOUSE PROJECT PART 1
BY LESLIE MILBURN

THIS PROJECT IS AN ALTERNATIVE TO THE MOUSE PROJECT PRESENTED IN THE LAST JOURNAL AND ALLOWS A MOUSE TO BE CONNECTED WITHOUT MODIFICATIONS TO THE VZ. ALL MICE WORK ROUGHLY THE SAME WAY SO I SHALL NOT REPEAT HERE WHAT WAS DESCRIBED IN GARY BULLEY'S ARTICLE. THERE ARE TWO TYPES OF MOUSE:-

(1) A SERIAL MOUSE - THIS IS THE MOST COMMON TYPE OF MOUSE AND HAS AN RS232C INTERFACE. THIS CAN HAVE EITHER A DB9 OR DB25 CONNECTOR. THIS IS THE TYPE OF MOUSE WHICH CAN BE MODIFIED FOR THE VZ USING GARY'S METHOD. AN ALTERNATIVE SOLUTION IS TO BUILD AN RS232 PORT FOR THE VZ AND PLUG IT IN DIRECTLY.

(2) A BUS MOUSE - THIS TYPE OF MOUSE IS USED BY THE AMIGA, AMSTRAD, ATARI AND SOME IBM CLONES SUCH AS THE COMMODORE PC COLT, PC10 & PC20 AND NOW THE VZ. THE PLUG ON THIS TYPE OF MOUSE IS A DB9 PLUG. FOR THOSE WHO NEED HELP FINDING A SUITABLE MOUSE THERE IS ONE AVAILABLE FOR $50-00 (APPROX) FROM GRACE BROS. IT IS A QTRONIX MOUSE FOR THE AMIGA 500/1000/2000, SERIAL NUMBER X30B (BUS VERSION). HOWEVER, SHOP AROUND AS I'M SURE THAT YOU CAN BUY THEM CHEAPER.

FOR EACH OF THESE MICE, A TWO BUTTON AND THREE BUTTON VERSION EXISTS.

THE HARDWARE:-

TO CONNECT A BUS MOUSE TO THE VZ, THE 8 BIT INPUT PORT AS DESCRIBED IN JOURNAL #30 IS REQUIRED. THE PIN DEFINITIONS ARE AS FOLLOWS:-

PIN - DESCRIPTION	DATA LINE
1 - VERTICAL MOTION	- D0
2 - HORIZONTAL MOTION	- D1
3 VERTICAL MOTION 90 DEG. OUT OF PHASE OF 1.	- D2
4 - HORIZONTAL MOTION 90 DEG. OUT OF PHASE OF 2.	- D3
5 - MOUSE BUTTON 3 (OPTIONAL)	- D4
6 - MOUSE BUTTON 1	- D5
7 - +5 VOLTS	- N/A
8 - GROUND	- N/A
9 - MOUSE BUTTON 2	- D6

AS CAN BE SEEN ABOVE, SOME MODIFICATIONS TO THE INPUT PORT MUST BE MADE. THE CHANGES I MADE WERE TO USE A DB9 SOCKET AND CONNECT THE PINS AS SHOWN ABOVE. D7 MUST BE CONNECTED TO GROUND, THIS IS REQUIRED SO THAT THE MOUSE DRIVER CAN DETECT THE PRESENCE OF THE MOUSE PORT.

WARNING: WHEN CONNECTING THE 8 BIT INPUT PORT TO THE PRINTER INTERFACE, CHECK WHICH VERSION OF THE INTERFACE YOU HAVE. THIS IS BECAUSE SOME INTERFACES USE A4 TO TRIGGER THE 74LS138 NOT A0.

THIS MEANS THAT ALTHOUGH THE PRINTER IS ACCESSED WHEN A4 IS LOW (I.E. PORTS 0 - 15), THE INPUT PORT IS ACCESSED WHEN A4 IS HIGH (I.E. PORTS 16 - 31). THIS CAUSES ADDRESS CONFLICT WITH THE DISK SYSTEM. ALSO NOTE THAT THIS VERSION OF THE INTERFACE IS NOT IN THE VZ TECHNICAL MANUAL. BE CAREFUL!!

THE MOUSE DRIVER:-

THE VZ MOUSE DRIVER IS AN INTERRUPT DRIVEN MACHINE CODE PROGRAM WHICH INTERPRETS THE RAW DATA SENT TO THE VZ BY THE MOUSE. THIS RAW DATA INDICATES MOUSE MOVEMENT AND BUTTON PRESSES.

THE MAIN AIM OF THIS PROJECT IS TO PROVIDE MOUSE FACILITIES EQUIVALENT TO THOSE ON AN IBM PC. THE LATEST MICROSOFT MOUSE DRIVER (VERSION 8.0) PROVIDES 53 FUNCTION CALLS THAT LET PROGRAMS PERFORM SPECIFIC TASKS, SUCH AS CHECKING A MOUSE BUTTON. THE VZ MOUSE DRIVER PROVIDES ALMOST THE WHOLE SUITE OF MOUSE FUNCTIONS.

THE MOUSE DRIVER CAN BE BROKEN INTO THREE DISTINCT PARTS:-

(1) THE INTERRUPT ROUTINE,
(2) THE MOUSE FUNCTION LIBRARY,
(3) THE BASIC MOUSE FUNCTION.

NOTE:

(I) VERSION 1.03 OF THE VZ MOUSE DRIVER ONLY SUPPORTS TEXT MODE(0).
(II) THIS VERSION OF THE MOUSE DRIVER SUPPORTS A MOUSE CURSOR WHICH IS INDEPENDENT OF THE KEYBOARD CURSOR.

THE INTERRUPT ROUTINE:-

EVERYTIME AN INTERRUPT OCCURS THIS ROUTINE MUST BE CALLED. IT LOOKS AFTER THE POLLING OF THE MOUSE AND MAINTENANCE OF ALL MOUSE "VARIABLES". THE MORE TIMES THE MOUSE IS POLLED, THE MORE SENSITIVE THE MOUSE CURSOR MOVEMENT. THE NUMBER OF TIMES THE MOUSE IS POLLED CAN BE SET BY A PROGRAM.

THE MOUSE FUNCTION LIBRARY:-

AS MENTIONED ABOVE, THERE ARE 53 IBM MOUSE FUNCTIONS. NOT ALL FUNCTIONS ARE PROVIDED IN V1.03 OF THIS DRIVER. THESE FUNCTIONS ARE DOCUMENTED IN TWO READILY AVAILABLE BOOKS:-

(1) THE MICROSOFT MOUSE PROGRAMMERS REFERENCE - $65-00 FROM DYMOCKS
(2) THE DOS PROGRAMMERS REFERENCE - $60-00 (APPROX) ALSO FROM DYMOCKS.

IN THE NEXT JOURNAL I SHALL DESCRIBE EACH OF THE AVAILABLE FUNCTIONS IN DETAIL.

THE BASIC MOUSE FUNCTION:-

TO ALLOW THE BASIC PROGRAMMER EASY ACCESS TO THE MOUSE FUNCTION LIBRARY, A NEW BASIC COMMAND HAS BEEN PROVIDED. THE SYNTAX OF THIS COMMAND IS:-

MOUSE(M1%, M2%, M3%, M4%).

M1% IS THE MOUSE FUNCTION NUMBER.
M2% - M4% ARE PARAMETERS TO THE MOUSE FUNCTIONS.

NOTE: NOT ALL PARAMETERS ARE USED BY ALL MOUSE FUNCTIONS BUT THEY MUST STILL BE PROVIDED IN THE CALL.

THE NEW COMMAND MAY BE CALLED FROM WITHIN A BASIC PROGRAM OR IN IMMEDIATE MODE. ALTHOUGH YOU MAY USE INTEGER CONSTANTS AND VARIABLES TO CALL MOUSE FUNCTIONS, RETURN VALUES ARE PASSED BACK TO THE PROGRAM. THESE RETURN VALUES WILL ALWAYS BE STORED IN THE VARIABLES M1%, M2%, M3% AND M4%, AND AS SUCH PROGRAMS SHOULD AVOID STORING DATA IN THESE VARIABLES.

TO BE CONTINUED .....

```
00001 ;****************************
00002 ;THIS PROGRAM IS A SAMPLE
00003 ;USER FUNCTION TO BE USED
00004 ;IN CONJUNCTION WITH KSCAN
00005 ;****************************
00006 BOX  EQU  28866
00007 SCAN EQU  002BH
00008 PWON EQU  4008H
00009 PWOF EQU  400BH
00010 SRCH EQU  402CH
00011 LOAD EQU  4041H
00012 FTYP EQU  "T"
00013 NEW  EQU  1B4DH
00014 RUN  EQU  36E9H
00015 EXIT EQU  1A19H
00016 WLTC EQU  6800H
00017 RLTC EQU  30779
00018 ;
00019 ;****************************
00020 ;USER: APPLICATION PROVIDED
00021 ;      USER FUNCTION.
00022 ;
00023 USER PUSH AF
00024      CP   2
00025      JR   NZ,USE4
00026      CALL SAVE
00027      CALL DBOX
00028      LD   HL,MES1
00029 USE1 CALL MSG
00030 USE2 CALL SCAN
00031      CP   "N"
00032      JR   Z,USE3
00033      CP   "Y"
00034      JR   NZ,USE2
00035 ;LOAD THE DISK FILENAME
00036      CALL LNAM
00037 ;CHECK IF THE FILE EXISTS
00038      CALL FIND
00039      LD   HL,MES2
00040      CP   2
00041      JR   NZ,USE1
00042 ;CLEAR THE PROGRAM AREA
00043      CALL NEW
00044      CALL PWON
00045      CALL LOAD
00046      PUSH AF
00047      CALL PWOF
00048      POP  AF
00049      LD   HL,MES3
00050      CP   0
00051      JR   NZ,USE1
00052      LD   DE,(30884)
00053      JP   RUN
00054 USE3 CALL REST
00055      POP  AF
00056      XOR  A
00057      RET
00058 USE4 POP  AF
00059      RET
00060 ;****************************
00061 ;DBOX: DRAW THE POPUP MENU.
00062 ;
00063 DBOX LD   HL,BOX
00064      LD   (HL),142
00065      INC  HL
00066      LD   (HL),140
00067      LD   DE,BOX+2
00068      LD   BC,26
00069      LDIR
00070      LD   (HL),141
00071      LD   HL,BOX+64
00072      LD   (HL),139
00073      INC  HL
00074      LD   (HL),131
00075      LD   DE,BOX+66
00076      LD   BC,26
00077      LDIR
00078      LD   (HL),135
00079      RET
00080 ;****************************
00081 ;MSG: DISPLAY USER MESSAGE.
00082 ;
00083 MSG  LD   DE,BOX+32
00084      LD   BC,28
00085      LDIR
00086      RET
00087 MES1 EQU  $
00088 :3
00089      DEFB 138
00090 *GO TO THE MAIN MENU (Y/N)?*
00091      DEFB 133
00092 MES2 EQU  $
00093 :3
00094      DEFB 138
00095 *ERROR: CHANGED DISK (Y/N)?*
00096      DEFB 133
00097 MES3 EQU  $
00098 :3
00099      DEFB 138
00100 * DISK ERROR: RETRY (Y/N)? *
00101      DEFB 133
00102 ;****************************
00103 ;SAVE: SAVE THE SCREEN INFO
00104 ;      WHERE THE POP UP IS
00105 ;      TO BE DISPLAYED.
00106 ;
00107 SAVE LD   HL,BOX
00108      LD   DE,BOX1
00109      LD   BC,96
00110      LDIR
00111      LD   A,32
00112      LD   (WLTC),A
00113      RET
00114 BOX1 DEFS 100
00115 ;****************************
00116 ;REST: THIS RESTORES THE
00117 ;      DISPLAY BACK TO ITS
00118 ;      ORIGINAL STATE.
00119 ;
00120 REST LD   HL,BOX1
00121      LD   DE,BOX
00122      LD   BC,96
00123      LDIR
00124      LD   A,(RLTC)
```

```
00125        LD    (WLTC),A
00126        RET
00127 ;****************************
00128 ;LNAM: LOAD THE NAME OF THE
00129 ;      PROGRAM INTO THE DISK
00130 ;      FILENAME BUFFER.
00131 ;
00132 LNAM LD    HL,FNAM
00133      PUSH IY
00134      POP  DE
00135      INC  DE
00136      LD   BC,8
00137      LDIR
00138      LD   A,FTYP
00139      LD   (DE),A
00140      RET
00141 FNAM EQU  $
00142 :1
00143 *MAINMENU*
00144 ;****************************
00145 ;FIND: CHECK THAT THE DISK
00146 ;      CONTAINS THE PROGRAM
00147 ;      TO LOAD.
00148 ;
00149 FIND CALL PWON
00150      CALL SRCH
00151      PUSH AF
00152      CALL PWOF
00153      POP  AF
00154      RET
```

EDITOR'S COMMENTS:

THE SOURCE CODE FOR KSCAN2.1 CONSISTS OF 5 FILES, INIT0, BASIC, SYSTEM, PLOT AND MATRIX. BELOW YOU WILL FIND DETAILS OF EACH FILE TOGETHER WITH NUMBER OF LINES OF EACH FILE AND NUMBER OF PAGES PER FILE WHEN PRINTED IN TWO COLUMNS PER PAGE (SEE PAGE 6 FOR EXAMPLE).

IF I PRINTED 3/4 PAGES PER ISSUE IT WOULD TAKE 4/5 ISSUES TO COMPLETE WHICH IS TOTALLY UN-ACCEPTABLE.

ONLY ONE EDITOR ASSEMBLER IS CAPABLE OF ASSEMBLING THE SOURCE CODE, LESLIE MILBURN'S DISKOPS6 (64K VERSION - PUBLIC DOMAIN) MUST BE USED. BESIDES NEEDING THE EXTRA MEMORY DISKOPS6 ALSO HAS ADDITIONAL COMMANDS, ONE OF WHICH IS USED TO 'LINK' THE SOURCE CODE TOGETHER.

```
INIT0    13 05 A813 DEEF 36DC - LINK BASIC  - 694 LINES -  5.60 PAGES
BASIC    1A 05 A813 C831 201E - LINK SYSTEM - 451 LINES -  3.63 PAGES
SYSTEM   0B 0C A813 B7E7 0FD4 - LINK PLOT   - 180 LINES -  1.45 PAGES
PLOT     0D 0D A813 AA8A 0277 - LINK MATRIX -  21 LINES -  0.16 PAGES
MATRIX   0E 03 A813 D04D 283A -             - 599 LINES -  4.83 PAGES
                                            ------------------------
                                            1945 LINES - 15.68 PAGES

USER     02 00 A813 B34B 0B38 - PRINTED THIS ISSUE.
KSCAN2.1 01 00 7AE9 8267 077E - KSCAN2.1 OBJECT CODE (M/C)
```

FOR ANY INFORMATION ON KSCAN2.1, ITS SOURCE CODE OR DISKOPS6 CONTACT LESLIE MILBURN. ADDRESS AND PHONE NUMBER ON PAGE 12.

EXT DOS V12.1 COMMANDS

FORMAT

THIS IS THE SAME AS DOS INIT. IT INITIALISES A DISK FOR USE. WHEN FINISHED IT PROMPTS USER FOR A DISK LABEL AND THE CURRENT DATE.

LABEL - THIS ALLOWS THE USER TO PUT A LABEL ON A FORMATTED DISK.

VOL - THIS ALLOWS THE USER TO READ BACK THE DISK LABEL.

NOTE: THE DISK LABEL USES A S:FILETYPE (SYSTEM FILE) AND OCCUPIES ONE SECTOR ON THE FLOPPY DISKETTE. IF YOU WISH TO PLACE THE DISK LABEL ON PART OF THE DISK WHICH IS NOT USED THEN I SUGGEST THAT THE SECOND HALF OF TRACK 0 SECTOR 15 BE USED. THIS IS CURRENTLY UNUSED.

DIS? - (?) = FILETYPE (OPTIONAL)

THIS DISPLAYS ALL FILES ON THE FLOPPY DISKETTE BY ?:FILETYPE. THE (?) IS OPTIONAL BUT IF NOT SPECIFIED THEN ALL FILES WILL BE DISPLAYED. IF T IS SPECIFIED, (IE. DIST) THEN ONLY T:FILETYPES WILL BE DISPLAYED. THE OUTPUT FORMAT IS AS FOLLOWS:

```
FILETYPE:FILENAME -    START     - MEMORY ADDRESS
                     TRACK SECTOR -  START END
       T:FILENAME      01 00         7AE9 8673
```

AFTER EACH SECTOR OF THE DISK DIRECTORY HAS BEEN DISPLAYED THE OUTPUT WILL BE PAUSED. PRESS SPACE TO CONTINUE LISTING OR BREAK TO EXIT.

CHA"FILENAME",? - (?) = FILETYPE

THIS FACILITY ALLOWS YOU TO CHANGE THE FILETYPE OF A FILE ON DISK. THE FILENAME AND FILETYPE ARE REQUIRED.

NOTE: AFTER CHANGING THE FILETYPE THE PROGRAM OR DATA MAY NO LONGER RUN OR LOAD. CHA CAN BE USED TO PROTECT FILES AGAINST ACCIDENTAL ERASURE OR TO CONVERT SOURCE CODE FILES FOR USE WITH OTHER EDITOR ASSEMBLERS.

REL"FILENAME",XXXX

THIS FACILITY ALLOWS YOU TO RELOCATE A FILE. THE FILENAME AND NEW START ADDRESS IS REQUIRED. A NEW END ADDRESS IS AUTOMATICALLY CALCULATED AND BOTH THE NEW START AND END ADDRESSES ARE PLACED INTO THE DISK DIRECTORY. USE HEX NUMBERS ONLY.

NOTE: A BASIC PROGRAM CAN BE RELOCATED TO A NEW START ADDRESS. HOWEVER, SOMETIMES THE RUN COMMAND WILL NOT WORK. HINT: TRY TYPING THE FIRST LINE IN IMMEDIATE MODE THEN GOTO 20 (FOR EXAMPLE).

EXA,XXXX,YYYY

THIS FACILITY ALLOWS YOU TO EXAMINE ANY TRACK (XXXX) AND SECTOR (YYYY) ON THE FLOPPY DISKETTE. USE ONLY HEX NUMBERS. THE SECTOR WILL THEN BE DISPLAYED. PRESS SPACE TO CONTINUE SEEING THE FOLLOWING SECTORS ON THE DISKETTE. TO EXIT USE BREAK.

```
00001 ;****************************
00002 ;EXTENDED DOS COMMANDS V12.1
00003 ;
00004 ;CODE IS NOT RELOCATABLE.
00005 ;****************************
00006 PCHR EQU  033AH
00007 SYER EQU  1997H
00008 GCHR EQU  1D78H
00009 PMSG EQU  2B75H
00010 PWON EQU  4008H
00011 PWOF EQU  400BH
00012 DERR EQU  400EH
00013 SRCH EQU  402CH
00014 SECT EQU  4035H
00015 CSI  EQU  401DH
00016 WRTE EQU  4032H
00017 INBF EQU  79E8H
00018 INPT EQU  0567H
00019 FSAV EQU  4044H
00020 COMA EQU  44
00021 HEX  EQU  4020H
00022 CLS  EQU  01C9H
00023 WKEY EQU  49H
00024 BAS  EQU  1A19H
00025 NEW  EQU  1B4DH
00026 RST  EQU  7804H
00027 ORG  EQU  7AE9H
00028 SBAS EQU  78A4H
00029 TOM  EQU  78B1H
00030 HTAB EQU  78A6H
00031 SSPC EQU  78A0H
00032 ;
00033 ;****************************
00034 ;BRUN: THIS FUNCTION IS
00035 ;      CALLED WHEN RUN FROM
00036 ;      DISK.
00037      CALL CLS
00038 BRUN LD   SP,7700H
00039      LD   HL,(RST)
00040      LD   (VCTR),HL
00041      LD   HL,CMD
00042      LD   (RST),HL
00043      DEC  HL
00044      LD   (TOM),HL
00045      LD   DE,50
00046      OR   A
00047      SBC  HL,DE
00048      LD   (SSPC),HL
00049      LD   HL,ORG
00050      LD   (SBAS),HL
00051      CALL NEW
00052      JP   BAS
00053 ;****************************
00054 ;CMD: THIS FUNCTION CHECKS
00055 ;     AND PROCESSES THE NEW
00056 ;     COMMANDS.
00057 CMD  EXX
00058      LD   HL,1D5BH
00059      POP  DE
00060      OR   A
00061      SBC  HL,DE
00062      PUSH DE
00063      EXX
00064      JP   NZ,GCHR
00065      PUSH HL
00066      CALL GCHR
00067      JR   NZ,CMD1
00068 POP  POP  HL
00069      LD   DE,(VCTR)
00070      PUSH DE
00071      RET
00072 CMD1 CP   "V"
00073      JR   NZ,CMD2
00074      CALL VOL
00075      JR   C,POP
00076      JR   CMDX
00077 CMD2 CP   "L"
00078      JR   NZ,CMD3
00079      CALL LABL
00080      JR   C,POP
00081      JR   CMDX
00082 CMD3 CP   "E"
00083      JR   NZ,CMD4
00084      CALL EXA
00085      JR   C,POP
00086      JR   CMDX
00087 CMD4 CP   "D"
00088      JR   NZ,CMD5
00089      CALL DIS
00090      JR   C,POP
00091      JR   CMDX
00092 CMD5 CP   "C"
00093      JR   NZ,CMD6
00094      CALL CHA
00095      JR   C,POP
00096      JR   CMDX
00097 CMD6 CP   "R"
00098      JR   NZ,POP
00099      CALL REL
00100      JR   C,POP
00101 CMDX POP  DE
00102      POP  DE
00103      LD   DE,1D1EH
00104      PUSH DE
00105      RET
00106 VCTR DEFW 0
00107 FNAM EQU  $
00108 :1
00109 *DISKINFO*
00110 ;****************************
00111 ;VOL: THIS DISPLAYS THE DISK
00112 ;     LABEL.
00113 ;
00114 VOL  CALL GCHR
00115      CP   "O"
00116      SCF
00117      RET  NZ
00118      CALL GCHR
00119      CP   "L"
00120      SCF
00121      RET  NZ
00122      CALL GCHR
00123      JP   NZ,SYER
00124 ;SAVE THE BUFFER POINTER
```

```
00125        PUSH HL
00126 ;LOAD THE FILENAME
00127        CALL LNAM
00128        DI
00129        CALL PWON
00130        CALL SRCH
00131        CP   2
00132        LD   A,13
00133        JP   NZ,DERR
00134 ;UPON EXIT FROM SEARCH DE
00135 ;POINTS TO START TRACK IN
00136 ;DIR ENTRY.
00137        LD   A,(DE)
00138        LD   (IY+18),A
00139        INC  DE
00140        LD   A,(DE)
00141        LD   (IY+17),A
00142        CALL SECT
00143        CALL PWOF
00144        EI
00145 ;CLEAR TWO SCREEN LINES.
00146        LD   HL,(7820H)
00147        PUSH HL
00148        LD   B,63
00149        LD   A,32
00150 VOL1   CALL PCHR
00151        DJNZ VOL1
00152        POP  HL
00153        LD   (7820H),HL
00154        LD   A,0
00155        LD   (HTAB),A
00156 ;GET POINTER TO DISK LABEL
00157        LD   H,(IY+50)
00158        LD   L,(IY+49)
00159 ;SAVE IT
00160        PUSH HL
00161 ;PLACE A NULL BYTE AFTER
00162 ;EACH STRING. THIS MEANS
00163 ;THAT DISK LABELS CREATED BY
00164 ;QUICKWRITE CAN BE DISPLAYED
00165        PUSH HL
00166        LD   DE,62
00167        ADD  HL,DE
00168        LD   (HL),0
00169        ADD  HL,DE
00170        INC  HL
00171        INC  HL
00172        ADD  HL,DE
00173        LD   (HL),0
00174 ;DISPLAY DISK LABEL.
00175        POP  HL
00176        CALL PMSG
00177        LD   A,13
00178        CALL PCHR
00179        POP  HL
00180        LD   DE,63
00181        ADD  HL,DE
00182        CALL PMSG
00183        LD   A,13
00184        CALL PCHR
00185        POP  HL
00186        OR   A
00187        RET
00188 ;****************************
00189 ;LNAM: THIS FUNCTION LOADS
00190 ;      THE FILENAME INTO THE
00191 ;      DISK NAME BUFFER.
00192 LNAM LD   HL,FNAM
00193        PUSH IY
00194        POP  DE
00195        INC  DE
00196        LD   BC,8
00197        LDIR
00198        LD   (IY+9),"S"
00199        RET
00200 ;****************************
00201 ;LABL: THIS FUNCTION PLACES
00202 ;      A LABEL ON A DISK.
00203 LABL CALL GCHR
00204        CP   "A"
00205        SCF
00206        RET  NZ
00207        CALL GCHR
00208        CP   "B"
00209        SCF
00210        RET  NZ
00211        CALL GCHR
00212        CP   "E"
00213        SCF
00214        RET  NZ
00215        CALL GCHR
00216        CP   "L"
00217        SCF
00218        RET  NZ
00219        CALL GCHR
00220        JP   NZ,SYER
00221 ;SAVE BUFFER POINTER.
00222        PUSH HL
00223 ;SAVE INPUT BUFFER CONTENTS.
00224        LD   HL,INBF
00225        LD   DE,LBF2
00226        LD   BC,64
00227        LDIR
00228        LD   HL,MES1
00229        CALL PMSG
00230 LAB1   CALL INPT
00231        JP   C,LABX
00232        LD   A,(INBF)
00233        CP   32
00234        JR   Z,LAB1
00235 ;COPY INPUT TO INTERNAL
00236 ;BUFFER.
00237        LD   DE,LBF1
00238        LD   HL,INBF
00239        LD   BC,62
00240        LDIR
00241        LD   HL,MES2
00242        CALL PMSG
00243 LAB2   CALL INPT
00244        JP   C,LABX
00245        LD   A,(INBF)
00246        CP   32
00247        JR   Z,LAB2
00248 ;COPY INPUT TO INTERNAL
```

```
00249 ;BUFFER.
00250       LD    DE,LBF1+63
00251       LD    HL,INBF
00252       LD    BC,62
00253       LDIR
00254 ;PLACE LABEL NAME INTO
00255 ;DISK BUFFER.
00256       CALL LNAM
00257       DI
00258       CALL PWON
00259 ;CHECK IF THE FILE ALREADY
00260 ;EXISTS.
00261       CALL SRCH
00262       CP    2
00263       JP    Z,DERR
00264 ;SAVE THE CURRENT START AND
00265 ;END OF BASIC POINTERS.
00266       LD    HL,(78A4H)
00267       PUSH HL
00268       LD    HL,(78F9H)
00269       PUSH HL
00270 ;SET THE START AND END
00271 ;ADDRESS OF LABEL DATA.
00272       LD    HL,LBF1
00273       LD    (78A4H),HL
00274       LD    HL,LBF1+126
00275       LD    (78F9H),HL
00276       CALL FSAV
00277       CALL PWOF
00278       EI
00279 ;RESTORE ORIGINAL START AND
00280 ;END OF BASIC POINTERS.
00281       POP   HL
00282       LD    (78F9H),HL
00283       POP   HL
00284       LD    (78A4H),HL
00285 LABX LD    HL,LBF2
00286 ;RESTORE INPUT BUFFER CONTS.
00287       LD    DE,INBF
00288       LD    BC,64
00289       LDIR
00290       POP   HL
00291       OR    A
00292       RET
00293 MES1 EQU   $
00294 :1
00295 *ENTER DISK LABEL:-*
00296       DEFB 13
00297       DEFB 0
00298 MES2 EQU   $
00299 *ENTER TODAY'S DATE:-*
00300       DEFB 13
00301       DEFB 0
00302 ;INTERNAL BUFFER FOR STORING
00303 ;THE DISK LABEL.
00304 LBF1 DEFS 126
00305 LBF2 DEFS 64
00306 ;*****************************
00307 ;EXA: THIS FUNCTION ALLOWS
00308 ;      ANY SECTOR TO BE
00309 ;      VIEWED.
00310 EXA  CALL GCHR
00311       CP    "X"
00312       SCF
00313       RET   NZ
00314       CALL GCHR
00315       CP    "A"
00316       SCF
00317       RET   NZ
00318       CALL GCHR
00319       CP    COMA
00320       SCF
00321       RET   NZ
00322 ;INCREMENT POINTER PAST THE
00323 ;COMMA.
00324       INC   HL
00325       CALL HEX
00326       RET   C
00327       PUSH HL
00328       LD    HL,39
00329       RST   18H
00330       POP   HL
00331       RET   C
00332       LD    (IY+18),E
00333 ;CHECK NEXT CHAR.
00334       LD    A,(HL)
00335       CP    COMA
00336       SCF
00337       RET   NZ
00338       INC   HL
00339       CALL HEX
00340       RET   C
00341       PUSH HL
00342       LD    HL,16
00343       RST   18H
00344       POP   HL
00345       RET   C
00346       LD    (IY+17),E
00347       DEC   HL
00348       CALL GCHR
00349       JP    NZ,SYER
00350 ;SAVE BUFFER POINTER
00351       PUSH HL
00352       CALL CLS
00353 EXA1 DI
00354       CALL PWON
00355       CALL SECT
00356       CALL PWOF
00357       EI
00358 ;DISPLAY THE SECTOR.
00359       CALL DSCT
00360       LD    A,(IY+17)
00361       INC   A
00362       CP    16
00363       JR    Z,EXA4
00364 EXA2 LD    (IY+17),A
00365 EXA3 CALL WKEY
00366       CP    1
00367       JR    Z,EXAX
00368       CP    32
00369       JR    Z,EXA1
00370       JR    EXA3
00371 EXA4 LD    A,(IY+18)
00372       INC   A
```

CONTINUED NEXT ISSUE . . .

DISKOPS1 & 2 (DISK OPERATIONS) 40/12

INSTRUCTIONS BY LESLIE MILBURN

THIS IS A MACHINE CODE UTILITY WHICH ALLOWS MACHINE CODE PROGRAMMERS USING THE DICK SMITH EDITOR ASSEMBLER TO SAVE AND LOAD THEIR SOURCE CODE TO AND FROM DISK. (NOTE TO SAVE OBJECT CODE A TAPE MUST STILL BE USED) AS THE PROGRAM IS NOT RELOCATABLE TWO VERSIONS ARE SUPPLIED ON DISK, THESE ARE:-

DISKOPS1 - EXPANDED VZ200
DISKOPS2 - EXPANDED VZ300

GETTING THE PROGRAMS OPERATIONAL:-

STEP 1: POKE 30897,200:POKE 30898,242
STEP 2: BRUN"EDITOR ASSEMBLER"
OR: CRUN"EDITOR ASSEMBLER"
STEP 3: ONCE THE EDITOR ASSEMBLER IS OPERATIONAL ENTER THE FOLLOWING LINE:-
JP 4005H - ASSEMBLE AND RUN. THIS WILL REBOOT THE DOS.
STEP 4: BRUN"DISKOPS 1 OR 2" DEPENDING ON THE MEMORY SIZE OF YOUR VZ.

THE ADDITIONAL COMMANDS ARE:-

TU:FILENAME - LOAD SOURCE FILE FROM DISK.
TR:FILENAME - SAVE SOURCE FILE ON DISK.
TD: - THIS IS FOR USE WHEN THE SOURCE FILE ISN'T COMPLETE. THE SOURCE CODE IS SAVED WITH THE FILENAME OF SOURCE. IF THE FILE ALREADY EXISTS THEN THE NEW FILE REPLACES THE FILE ON DISK.

WARNING:-

THE TD: ROUTINE INTERFERES WITH THE DISK TRACK MAP SO USE THIS WITH CAUTION. I RECOMMEND THAT A SEPARATE DISK BE USED AS IT IS POSSIBLE FOR DISK FILES TO BE DAMAGED.

NOTE:-

ALL SOURCE CODE FILES WHEN SAVED ARE OF FILETYPE 'W' AND ONLY THE FIRST 8 CHARACTERS OF THE FILENAME ARE RECOGNISED, IE: W:FILENAME

ERROR MESSAGES:-

IF THE ORIGIN IS NOT SET TO THE FIRST FREE BYTE AFTER THE SOURCE CODE USING THE 'O' <<RETURN>> COMMAND THEN YOU WILL GET A *** WRONG ORIGIN FOR RUN *** ERROR.

INFORMATION AND SOURCE CODE:-

WRITE TO LESLIE MILBURN:- FOR A SOURCE CODE LISTING OR IF YOU HAVE ANY OTHER PROBLEMS.

LESLIE MILBURN 11 SWORDFISH AVENUE RABY 2566 PHONE (02) 603-5981

A SOURCE CODE LISTING IS AVAILABLE FOR $5-00.

LISTING 5:

```
0010 '*************************
0020 '***  1536 VZ SCREENS  ***
0030 '***    VERSION 1.3    ***
0040 '***  R.B.K.  18/5/86  ***
0050 '*************************
0088 '
0089 '***FIND TOP OF MEMORY AND PROTECT 20 BYTES BY LOWERING TOM.
0090 IF PEEK (30791)=77 THEN GOTO 500       :'7847H AS LOAD CHECK.
0100 M1=PEEK(30898):L1=PEEK(30897)          :'TOM POINTERS.
0110 TM=M1*256+L1-21                        :'PROTECT 20 BYTES.
0120 MS=INT(TM/256):LS=TM-MS*256            :'NEW TOM POINTERS.
0130 POKE 30898,MS:POKE 30897,LS            :'SET NEW TOM.
0140 CLEAR 50                               :'RESET ALL POINTERS.
0150 'THIS ACTION DESTROYS THE VLT-ALL VARIABLES MUST BE RENEWED.
0160 M1=PEEK(30898):L1=PEEK(30897)          :'NEW TOM POINTERS.
0170 ST=M1*256+L1+1                         :'NEXT ADDR IN RESERVED
0180 MS=INT(ST/256):LS=ST-256*MS            :'START OF ROUTINE.
0190 IF ST>32767 THEN ST=ST-65536           :'CONVERT TO SIGNED INT
0198 '
0199 '***LOAD MACHINE CODE INTO PROTECTED TOM AND SET USR POINTER
0200 CLS:PRINT@230,"LOADING MACHINE CODE"
0210 CS%=0                                  :'CHECK SUM COUNTER.
0220 FOR ID=ST TO ST+15                     :'LOAD 16 BYTES TO TOM.
0230    READ VL%:POKE ID,VL%:CS%=CS%+VL%    :'LOAD M/L INTO PROTECT
0240 NEXT ID
0250 IF CS%<>1483 THEN GOTO 1000            :'WRONG DATA.
0260 POKE 30863,MS:POKE 30862,LS            :'SET USR POINTERS.
0270 POKE 30791,77                          :'SET LOAD CHECK.
0399 '
0400 '***Z-80 BLOCK MOVE ROUTINE.
0410 DATA 058,033,121   :'LD A,(7921H) PICK UP VALUE FROM USR().
0420 DATA 033,000,112   :'LD HL,7000H  (#28672D START VIDEO RAM)
0430 DATA 017,001,112   :'LD DE,7001H  (#28673D NEXT)
0440 DATA 001,255,007   :'LD BC,07FFH  (#2047D  SIZE OF VIDEO RAM)
0450 DATA 119           :'LD (HL),A
0460 DATA 237,176       :'LDIR         (BLOCK LOAD COMMAND)
0470 DATA 201           :'RET
0498 '
0499 '***INITIALIZE DELAYS - CONTROL SPEED OF EXECUTION BY D.
0500 T%=0                                   :'TONE IS REST (0-31)
0510 D%=4                                   :'DELAY IS SHORT(1-9)
0520 P%=30744                               :'SET UP SCREEN MODE.
0594 '
0595 '***THIS LOOP DOES A LOT OF WORK.
0596 '***IT LOADS ALL VALUES INTO THE A-REG FOR VIEWING.
0597 '***IT SWAPS BETWEEN HI- AND LO-RES SCREEN MODES.
0598 '***IT SWITCHES BETWEEN NORMAL AND INVERSE SCREEN TYPES.
0599 '***IT SWITCHES BETWEEN GREEN AND ORANGE BACKGROUNDS.
0600 FOR A%=0 TO 255            :'ALL VALUES PUT INTO A=REG.
0610    MODE(0):POKE P%,0
0620    CLS:PRINT@234,"VALUE = ";A%:SOUND T%,D%
0630    X=USR(A%)
0640    COLOR,0:SOUND T%,D%
0650    COLOR,1:SOUND T%,D%
0660    POKE P%,1
0670    X=USR(A%)
0680    COLOR,0:SOUND T%,D%
0690    COLOR,1:SOUND T%,D%
0699 '
```

```
0700 '  HI-RES SCREENS.
0710    MODE(1)
0720    X=USR(A%)
0730    COLOR,0:SOUND T%,D%
0740    COLOR,1:SOUND T%,D%
0750 NEXT A%
0800 GOTO 600
0998 '
0999 '***WRONG DATA MESSAGE.
1000 CLS:PRINT@230,"ERROR IN DATA ";CS%
1010 END
```

FUNCTIONS IN BASIC

IT MAY BE A SURPRISE TO MOST BASIC PROGRAMMERS BUT THE FUNCTION (FN) COMMAND, ALONG WITH THE SUBROUTINE (GOSUB) COMMAND, ARE PROBABLY THE MOST USEFUL BASIC COMMANDS. THEY ARE CONCISE AND CLARIFY BASIC CODING CONSIDERABLY. UNFORTUNATELY ONLY GOSUB IS SUPPORTED ON THE VZ.

I HAVE HAD A NUMBER OF QUERIES FROM USERS ON HOW TO USE THE FN STATEMENT - PARTICULARLY WHEN TRYING TO CONVERT PROGRAMS WRITTEN FOR OTHER MACHINES. IN THIS BRIEF NOTE, I WILL TRY TO EXPLAIN HOW TO GET AROUND IT.

IN THE LEVEL II BASIC THAT IS IN THE VZ'S ROM, TWO TYPES OF FUNCTION ARE SUPPORTED -

1. LIBRARY (OR SYSTEM) FUNCTIONS, AND
2. USER-DEFINED FUNCTIONS.

FUNCTIONS CAN BE USED TO MANIPULATE NUMERIC OR STRING DATA TYPES. THE VZ SUPPORTS A NUMBER OF INTRINSIC OR LIBRARY FUNCTIONS SUCH AS SQR, ATN, RND, CHR$, LEFT$ AND INT ETC. THE PROCEDURES FOR THESE ARE IMBEDDED IN THE ROM AS BASIC UTILITIES. STEVE OLNEY'S EXTENDED BASIC "WAKES UP" A FEW MORE, SUCH AS DEFINT, CSNG AND STRING$.

UNFORTUNATELY ONE OF THE OMISSIONS FROM THE FULL LEVEL II IMPLEMENTATION ON THE VZ IS THAT THE USER DEFINED FUNCTIONS (DEF FNX) ARE NOT SUPPORTED IN ANY WAY. RUSSELL HARRISON'S EXTENDED BASIC XB DOES PROVIDE SUPPORT FOR DEF FN HOWEVER. NOTE THAT FUNCTIONS CAN ONLY RETURN A SINGLE VALUE TO THE CALLING PROGRAM.

THE LACK OF THIS FEATURE OFTEN CROPS UP WHEN ATTEMPTING TO CONVERT PROGRAMS TO RUN ON THE VZ - BUT WRITTEN IN OTHER DIALECTS OF BASIC. THE CONCISE CODING INHERENT IN FUNCTION STATEMENTS IS ALSO A DESIRABLE FEATURE WHEN WRITING CLEAR CODE. FORTUNATELY A FAIRLY SIMPLE REMEDY IS AT HAND AND DESCRIBED BELOW.

THE FUNCTION STATEMENT HAS TWO COMPONENTS. THE FIRST IS THE DEFINITION OF THE FUNCTION, AND THE SECOND, IS THE ACTUAL IMPLEMENTATION OR CALL TO THAT DEFINITION. LET'S EXPLAIN.... SUPPOSE THAT WE WISH TO FREQUENTLY COMPUTE THE AREA OF A CIRCLE GIVEN A NUMBER OF VALUES FOR THE RADIUS. THE COMMAND LINE

```
10 DEF FNA(X) = 3.1416 * X * X
```

SHOULD BE DECLARED EARLY IN THE PROGRAM. DEF MEANS DEFINE, FNA MEANS FUNCTION A (ANY LETTER FROM A TO Z CAN BE USED.), AND (X) IS A DUMMY ARGUMENT FOR THE RADIUS USED BY THE FUNCTION. THE RIGHT HAND SIDE OF THE ASSIGNMENT IS THE EASILY RECOGNIZED FORMULA FOR CALCULATING THE AREA OF A CIRCLE.

LATER IN THE PROGRAM, THE VARIOUS VALUES FOR THE RADIUS ARE ASSIGNED TO VARIABLE V (EITHER FROM DATA OR INPUT STATEMENTS.) AND THEN THE FOLLOWING LINE MAY APPEAR -

```
210 PRINT V,FNA(V)
```

THE TWO LINES GIVEN (10 AND 210) WILL TOGETHER, PRINTOUT THE RADIUS AND THE CORRESPONDING AREA FOR A CIRCLE. THE CALL BY FNA SENDS THE INTERPETER OFF THROUGH THE BASIC PROGRAM TO FIND THE DEF FNA STATEMENT. WHEN LOCATED, THE VALUE V IS SUBSTITUTUED FOR THE DUMMY VARIABLE, THE CALCULATION CARRIED OUT, AND THE RESULT RETURNED TO THE CALLING SEQUENCE. (ONLY A SINGLE VALUE CAN BE RETURNED.)

AS ALREADY STATED, THIS NEAT CONSTRUCT DOES NOT EXIST IN VZ BASIC. THE APPROPRIATE USE OF THE GOSUB/RETURN COMMANDS CAN OVERCOME THIS SHORTFALL. THE GOSUB COMMAND CAN RETURN MANY VALUES - BUT A FEW MORE ASSIGNMENTS ARE REQUIRED BEFORE GOING TO THE SUBROUTINE AND YOU HAVE TO KEEP TRACK OF THE SUBROUTINES' LINE NUMBER.

AN EXAMPLE BEST ILLUSTRATES THIS - AND LET'S USE THE PREVIOUS EXAMPLE TO SHOW HOW IT CAN BE IMPLEMENTED ON THE VZ.....

```
10 INPUT "ENTER RADIUS OF CIRCLE ";R
20 GOSUB 1000
30 PRINT"RADIUS ";R,"AREA ";A,"CIRCUMFERENCE ";C
40 GOTO 10
1000 A=3.1416*R*R
1010 C=2*3.1416*R
1020 RETURN
2000 END
```

NOT TOO DIFFICULT TO SET UP IS IT? NOTICE THAT TWO VALUES (A AND C) WERE RETURNED. THE CODING AND PROGRAM FLOW IS NOT QUITE AS CLEAR AS USING THE FUNCTION STATEMENT.

HAVE FUN! AND DON'T BE FOXED BY FUNCTIONS WHEN NEXT CONVERTING PROGRAMS ONTO THE VZ.

TECHNICAL-DATA SHEET # 1

WORDPRO & DD POWER SUPPLY CIRCUITS:

BOTH CIRCUITS WERE SUPPLIED BY DAVE BOYCE SOME YEARS BACK AND ARE PUBLISHED AS A SUPPLEMENT TO THE TECHNICAL REFERENCE MANUAL.

WORDPRO CIRCUIT: - CAN SOMEONE HELP WITH A TRUTH TABLE FOR THE DECODING CIRCUIT PLEASE AS IT HAS SOME UNUSUAL DECODING, ED.

DD POWER SUPPLY CIRCUIT:

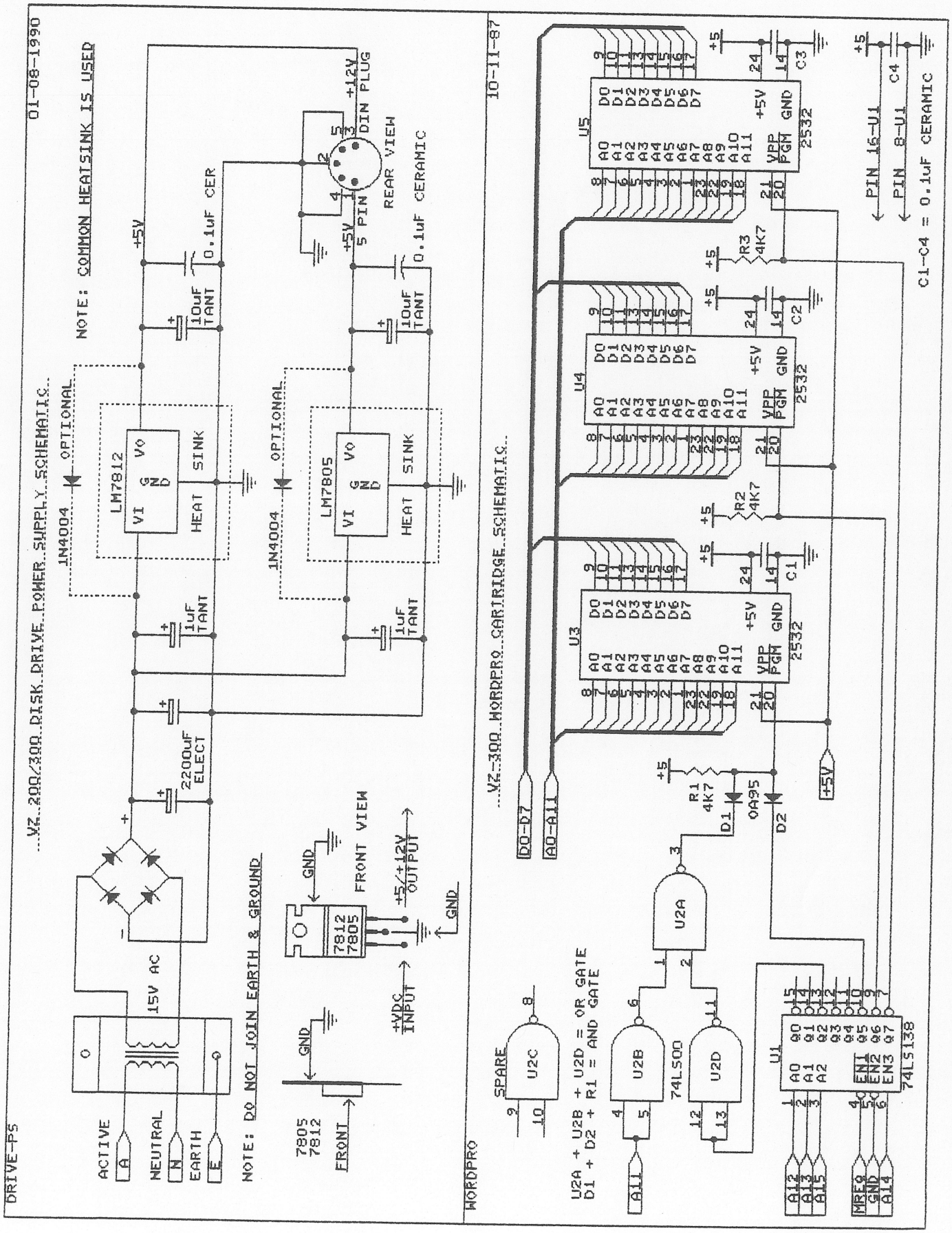
THIS IS A STANDARD POWER SUPPLY CIRCUIT WITH NOTHING FANCY. IT'LL WILL BE INVALUABLE FOR THOSE WITH A CROOK POWER SUPPLY UNIT OR FOR THOSE WHO NEED ONE AND WOULD LIKE TO BUILD ONE THEMSELVES NOW THAT THEY ARE NO LONGER AVAILABLE FROM DICK SMITH.

NOTE: IF THE DEMAND IS THERE I COULD PREPARE A PROJECT FOR PUBLICATION, BUT PLEASE LET ME KNOW AS SOON AS POSSIBLE, ED.

TECHNICAL DATA SHEET # 1

WORDPRO TAPE/DISK WORD PROC. CARTRIDGE CIRCUIT

DISK DRIVE POWER SUPPLY CIRCUIT



DRIVE HEAD PARK UPDATE

AS A TWO DRIVE USER I FOUND IF I USED PARK IN DRIVE 2 IT WOULD PARK DRIVE 2. ON POWER UP OR RESET DRIVE 1 WOULD BANG ITS HEAD OFF. JASON OAKLEY MODIFIED IT SO PARK WOULD ALWAYS PARK DRIVE 1 REGARDLESS OF WHICH DRIVE IT WAS RUN FROM.

LOAD PARK SOURCE CODE INTO YOUR EDITOR ASSEMBLER AND INSERT THE FOLLOWING LINES BY TYPING I19 AT THE COMMAND PROMPT.

```
020 ;SELECT DRIVE 1
021        LD   A,10H
022        LD   (IY+0BH),A
```

AS YOU INSERT THE LINES 020-022 THE OLD LINES FROM 020-070 ARE BEING RENUMBERED TO 023-073 TO END OF PROGRAM. NEXT EDIT LINES 063 AND 072 AS SHOWN BELOW.

```
063 *      PARK1 BY DAVE MITCHELL
072 *      DRIVE1 HEAD IS PARKED*
```

THAT FINISHES THE MODIFICATIONS. SAVE THE SOURCE AND OBJECT CODES AS SHOWN BELOW OR ACCORDING TO THE SYNTAX OF YOUR EDITOR ASSEMBLER.

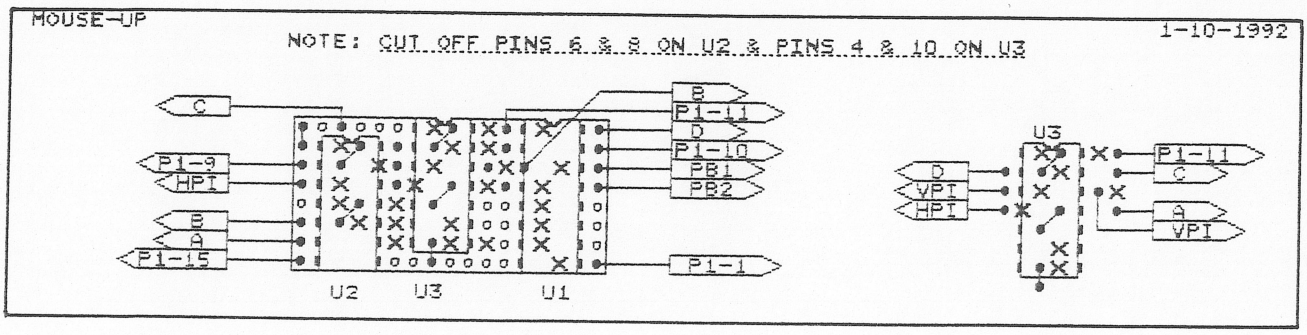
```
O7200H     - SET ORIGIN
A          - ASSEMBLE
TS:PARK-1  - SAVE SOURCE CODE
TO:PARK1   - SAVE OBJECT CODE
```

REGARDLESS OF WHICH DRIVE YOU BRUN"PARK1" FROM, DRIVE 1 WILL ALWAYS BE THE DRIVE WHICH IS PARKED, WHICH MEANS NO MORE UN-INTENTIONAL HEAD BANGING. HAPPY PARKING !!!

GARY BULLEY VZMOUSE ERRATTA:

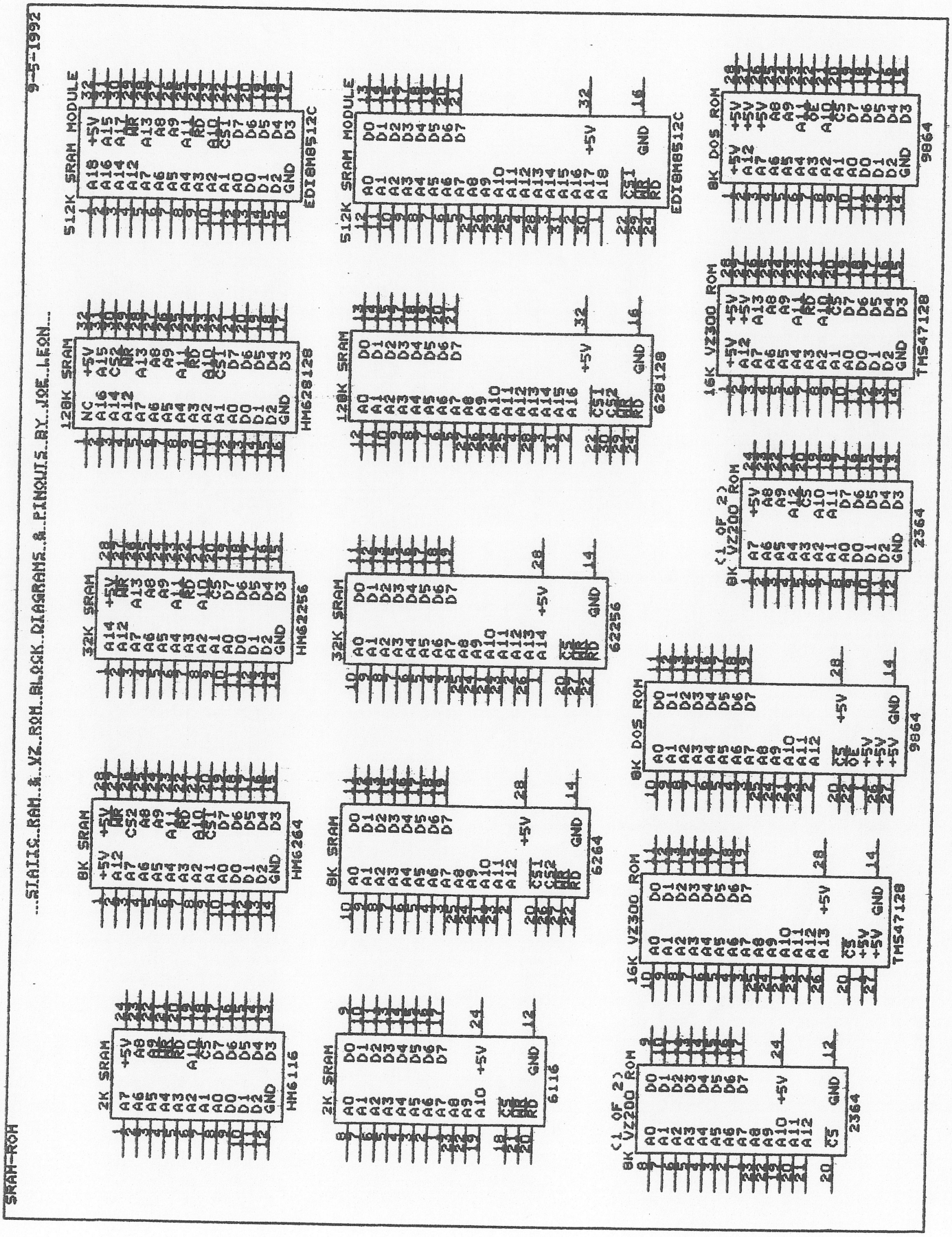
SOME ERRORS CREPT IN WHEN I WAS PREPARING GARY'S VZMOUSE PROJECT IN LAST ISSUE. MY APOLOGIES TO GARY BULLEY AND ALL CONCERNED, ED.

THE ERRORS WERE IN THE VERO BOARD LAYOUT AND CONNECTIONS. THE CORRECTED (I HOPE) VERSION IS SHOWN BELOW. DO NOT FORGET TO CUT OFF PINS 6 & 8 ON U2 AND PINS 4 & 10 ON U3 WHICH ARE NOT USED AS IT SIMPLIFIES BOARD LAYOUT. THE 'B' CONNECTION FROM PIN 5 ON U2 IS MADE DIRECTLY TO PIN 3 ON U1. OTHER CONNECTIONS ARE AS SHOWN.



STATIC RAMS & VZ ROMS

PINOUTS AND BLOCK DIAGRAMS ARE PRESENTED OF STATIC RAMS FROM 2K TO 128K, 512K MODULE, VZ200/300 8K & 16K BASIC ROMS AND 8K DOS ROM.



*** DAVE MITCHELL SOFTWARE ***

E & F WP PATCH 3.3: PATCH 3.3 WRITTEN BY DAVE MITCHELL WILL CONVERT YOUR E & F TAPE WORD PROCESSOR FOR FULL DISK USE WHILE RETAINING ALL ORIGINAL FUNCTIONS. IT ALSO HAS SHIFT LOCK AND PRINTER CONTROL CODES WHICH CAN BE IMBEDDED IN TEXT AND SAVED TO TAPE OR DISK.

BSTWP.F: THIS UTILITY PROVIDED WITH PATCH 3.3 WILL CONVERT BASIC PROGRAMS AND ED/ASS. SOURCE CODE FILES INTO WORD PROCESSOR FILES.

PRICE: AUS/NZ AU$20.00 - UPDATE - AUS-$10.00 - NZ-AU$11.00.

EXTENDED DOS V1.3: THESE COMMANDS ARE AT YOUR DISPOSAL: MERGE, DIRA, DIRA, DIRB, LDIRB, OLD, OLD., DEC, HEX, MENU, CODE, LTAB, MOVE AND UPDATE, STATUSA AND LSTATUSA. STATUSA AND LSTATUSA ALSO WORKS WITH VERSION 1.0 DOS

PRICE: AU$15.00 - POSTAGE INCLUDED

MENU/FILE COPIER: THIS UTILITY WILL READ YOUR DISK DIRECTORY AND PRESENT YOU WITH SEVERAL OPTIONS. USING THE CURSOR YOU CAN RUN/BRUN ANY PROGRAM OR SELECT FILE COPY, REN, ERASE, DRIVE 1 OR 2, ETC. BESIDES COPYING TEXT AND BINARY FILES ALL OTHER FILES CAN BE COPIED AS WELL EXEPT FOR DATA FILES.

PRICE: AU$15.00 - POSTAGE INCLUDED

FOR PURCHASE OR INFORMATION CONTACT:

DAVE MITCHELL 24 ELPHINSTONE STREET
NORTH ROCKHAMPTON QUEENSLAND 4701
AUSTRALIA - PHONE: (079) 27 8519

*** PETER HICKMAN SOFTWARE ***

VZ DISASSEMBLER: WHAT, ANOTHER DISASSEMBLER? BUT, YOU HAVE ALREADY GOT ONE? THIS ONE IS DIFFERENT! THIS PROGRAM IS ENTIRELY WRITTEN IN MACHINE CODE. IT ACTUALLY RUNS ABOUT 40 TIMES FASTER THAN D.S.E.'S DISASSEMBLER (OR ANY ONE ELSE'S). IT WILL DISASSEMBLE ANY PROGRAM THAT YOU CAN BLOAD INTO MEMORY. IT WORKS WITH ANY VZ CONFIGURATION. IT DISASSEMBLES EVEN THE 88 EXTRA Z80 OPCODES THAT ZILOG DOESN'T ADMIT TO.

PRICE: AU$25.00 - PRICE INCLUDES HARDCOPY MANUAL.
TAPE AND DISK VERIONS AVAILABLE.

VZ MODEM SOFWARE: DID YOU WANT TO TALK TO OTHER COMPUTERS VIA A MODEM? DID YOU BUY THE DSE TERMINAL EPROM, ONLY TO DISCOVER THAT IT ONLY WORKS WITH TAPE. IT ONLY ALLOWS YOU TO PRINT FILES, NOT SAVE THEM OR SEND THEM!

YOUR PROBLEMS ARE SOLVED! THE HICKMAN BROTHERS, PETER AND ANDREW, HAVE A BRAND NEW PROJECT WHICH WILL ALLOW YOU TO SEND, RECEIVE & SAVE FILES VIA A MODEM. IT WORKS WITH DISK!

SALE PRICE: $25.00 - INCLUDED ARE INSTRUCTIONS FOR THE HARDWARE MODIFICATIONS. A SMALL MODIFICATION IS NEEDED TO YOUR DISK CONTROLLER. YOUR USER GROUP MAY HELP YOU MODIFY YOUR COMPUTER TO USE THIS EXCITING NEW SOFTWARE!

THE MANUAL IS SUPPLIED ON DISK FOR PRINTING OUT WITH YOUR DISK VERSION OF E & F W/PROCESSOR. IF YOU DO NOT OWN AN E & F W/PROCESSOR THEN PLEASE ENCLOSE ANOTHER $5.00 (TOTAL $30.00) FOR PHOTOCOPYING AND POSTAGE OF THE MANUAL.

FOR PURCHASE OR INFORMATION CONTACT:
PETER HICKMAN PO BOX 8 WERRINGTON 2747

** ** CONTRIBUTIONS TO THE JOURNAL ** **

IF YOU ARE THINKING OF CONTRIBUTING TO THE JOURNAL THE PREFERED FORMAT IS BASIC LISTINGS, WORD PROCESSOR OR SOURCE CODE FILES ON TAPE OR DISK. FILES FROM THE FOLLOWING WORD PROCESSORS CAN BE ACCEPTED :-

E & F TAPE OR DISK PATCH 3.1-3.3, WORDPRO CARTRIDGE, WORDPRO PATCH, MOST SOURCE CODE FILES AND ALL QUICKWRITE WORD PROCESSOR FILES.

** ** CLUB MEETINGS - ALL WELCOME ** **

FIRST FRIDAY OF MONTH

** ** FUTURE MEETINGS - NEW VENUE ** **

AS MENTIONED BEFORE WE NO LONGER MEET AT JNC, BUT AT VARIOUS MEMBERS HOMES. MEETINGS WILL BE ONCE A MONTH AS BEFORE WITH THE DATES BEING FIRST FRIDAY OF THE MONTH.

BECAUSE OF SOME LOCAL MEMBERS HAVING TO WORK SHIFTWORK MEETING DATES WILL BE ADJUSTED TO ACCOMODATE THEM. WHETHER YOU ARE A LOCAL MEMBER, INTRA OR INTERSTATE VISITOR PLEASE CHECK WITH JOE LEON FIRST BEFORE COMING OUT.

JOE LEON 33 TIGHES TCE TIGHES HILL 2297 (049) 692 399

** CLUB COMMITTEE & SUBSCRIPTIONS **

PRESIDENT - ROSS WOODS - SECRETARY/EDITOR - JOE LEON
COMMITTEE MEMBERS - COLIN BRIDGE - PETER JONES

SUBSCRIPTION TO - AUST. - 3 ISSUES $11.00 - 6 ISSUES $21.00
H.V.VZ.JOURNAL - N. Z. - 3 ISSUES $13.00 - 6 ISSUES $26.00

FOR MORE INFORMATION CONTACT:

JOE LEON 33 TIGHES TCE TIGHES HILL 2297 (049) 692 399 AUSTRALIA

NOTE: PRICES INCLUDE POST & PACKING

** ** VZ USER GROUPS & PUBLICATIONS ** **

VZ DOWN UNDER - VZ MAGAZINE - 6 ISSUES - $18.00 PER ANUM
HARRY HUGGINS 12 THOMAS SREET MITCHAM VICTORIA 3132

WAVZ - WESTERN AUSTRALIA VZ USER GROUP
GRAEME BYWATER PO BOX 388 MORLEY W A 6062

BRISBANE VZ USERS WORKSHOP - C/O 63 TINGALPA ST. WYNUM WEST 4178
SOFTWARE FOR SALE - DISK MENU

SAPPHIRE PRODUCTIONS - VZ DISK MAGAZINE - PUBLIC DOMAIN
NOTE: VZ DISK MAGAZINE HAS CEASED PRODUCTION

NOTE: WHEN WRITING TO ANY ABOVE OR H.V.VZ. USERS' GROUP FOR INFORMATION PLEASE ENCLOSE A S.S.A.E. OR NZ 2 INT. REPLY COUPONS.